5.0 Documentation Assessment

A study sponsored by the SMART Project identified the documentation necessary to support the verification of mature models. The results of that study are published in a report entitled Software Verification Requirements Study (SVRS) for the SMART Project [Ref. 9]. The SVRS describes the minimum set of documents and content standards required to assist a potential user to evaluate the suitability of an existing model for a specific purpose and ensure that it has been rigorously verified against known standards and procedures. These documents should allow the potential user to: have confidence that the model is accurate; decide if the model simulates the problem(s) of concern; have sufficient information to install and run the program(s); modify the model to work on the target platform (if necessary); understand all inputs and outputs; and fix problems during model use, either due to runtime errors, incorrect input, or incorrect program operation.

The document used in formulating these recommendations was the Defense System Software Development, DOD-STD-2167A which established requirements to be met by government contractors for the acquisition, development, or support of software systems. This document grew out of the need to standardize and manage the development of computer software in the DoD community and it includes requirements for software documentation. After an extensive search for, and review of government requirements and guidelines, the following documents were identified as the minimum set necessary for mature model verification: Software User's Manual (SUM), Software Programmer's Manual (SPM), Software Analyst's Manual (SAM), Software Design Document (SDD), and Software Verification Report (SVR).

The SUM and SPM formats described in DOD-STD-2167A were tailored for digital simulation models. The SAM is not addressed by 2167A so its format was generated after a review of other sources. Electronic Combat Digital Evaluation System (ECDES) Model Documentation and Programming Guidelines were also used as guidelines for implementing DOD-STD-2167A. The existing SAMs for ESAMS, ALARM Version 3.0, and RADGUNS were also used to provide guidance for the recommendations. The SDD and SVR typically do not exist for mature models. However, SMART has sponsored tasks to support the development of equivalent documents for several models: the Conceptual Model Specification (CMS) and Verification Reports (VRs).

ALARM 3.1 was found to have three of the recommended documents: a SUM [Ref. 6], a SPM [Ref. 5], and a SAM [Ref. 3]. It should be noted that ALARM is a mature model that has evolved during the past 24 years. Like many models with similar histories, an SDD was never written. A SDD is necessary to conduct verification of the model since an SDD sets forth the requirements which are verified to have been correctly implemented. As part

of the SMART Project Verification and Validation (V&V) effort for ALARM, a SDD substitute has been written. This document, called the Post-Development Design Document for ALARM [Ref. 3], was developed and renamed as the

Conceptual Model Specification (CMS) upon incorporation into Accreditation Support Package (ASP-II). The SVR is a report of a verification effort. A substitute for the SVR, the Verification Report (VR), has been developed and is included in ASP-III. Updated CMS's and VRs will be written as the enhancement and verification of ALARM proceeds under the SMART Project. Table 5.0-1 summarizes the assessment of existing ALARM documentation.

Table 5.0-1 Documentation Assessment Summary for ALARM

Characteristic	SUM	SPM	SAM
Publication Date	June 1995	June 1995	June 1995
Applicability	ALARM 3.1	ALARM 3.1	ALARM 3.1
Completeness	Adequate (Complete except for error messages, assumptions and limitations, and descriptions of sample output files)	Adequate (Complete except for variable and common block descriptions, detailed module descriptions, and error diagnostics)	Adequate (Complete except for assumptions and limitations and detailed description of algorithms)
Compliance	Complies (Except for some minor modifications)	Complies (Except for some minor modifications)	Complies (Except for some minor modifications)

Note: The characteristics and adequacy of the model documentation are summarized in the above table using the following criteria:

* Completeness The completeness of the documentation is stated as "Complete," "Adequate," (the implication being incomplete, but adequate), "Inadequate," or "Nonexistent."

* ComplianceThe compliance of the documentation with referenced standards is stated as "Complies" or "Does Not Comply."

* ApplicabilityThe version of the model the documentation represents is stated as "Current" (the latest version) or "Version (n.n.n)."

5.1 COMPLETENESS

Tables 5.1-2 through 5.1-4 provide summaries of the completeness of the SUM, SPM, and SAM, detailed by elements required for each section. Summing the results out of a total of 43 content elements, 19 were included and complete, 19 were included but partially complete, and 5 were not included.

5.1.1 Completeness of Software User's Manual

The ALARM SUM is very close to providing all information required. Most of the information of the proposed SUM can be copied from the existing ALARM 3.1 SUM and only the format in which it is presented needs to be modified. The only serious deficiencies are the lack of an adequate list of error messages and a discussion of assumptions and limitations. Table 5.1-1 summarizes the contents of the ALARM SUM.

Table 5.1-1 Contents Summary of SUM for ALARM.

Software User's Manual	Complete
Title Page and Preliminary Information	P
1.1 Identification	Y
1.2 System Overview	Y
1.3 Document Overview	Y
2.0 Referenced Documents	Y
3.1 Initialization	Y
3.2 User Inputs	P
3.3 Links To Other Programs	Y
3.4 Outputs	P
4.0 Error Messages	P
5.0 Terms and Abbreviations	Y
Appendix A: Assumptions and Limitations	N

Notes:

Y Included and Complete

N Not Included

P Partial Treatment

5.1.2 Completeness of Software Programmer's Manual

The ALARM SPM is very close to providing all information required. Some of the information of the proposed SPM can be copied from the existing ALARM 3.1 SPM and only the format in which it is presented needs to be modified. Serious deficiencies include the lack of an adequate list of error messages, a high-level call hierarchy, a dictionary of variables, and the discussion of global variables (common blocks). Table 5.1-2 shows completeness for each standard SPM section.

Table 5.1-2 Contents Summary of SPM for ALARM.

Software Programmer's Manual	Complete
Title Page and Preliminary Information	P
1.1 Identification	Y
1.2 System Overview	Y-SPM, SUM
1.3 Document Overview	P
2.0 Referenced Documents	Y
3.1 Equipment Configuration	P
3.2 Operational Information	P
3.3 Compiling and Linking Instructions	P
4.1 Introduction to Programming Information	P
4.2 Call Hierarchy	P
4.3 Dictionary of Variables	P-SUM
4.4 Global Variables	N
4.5 Program, Subroutine, and Function Descriptions	P
4.6 Error Detection and Diagnostic Features	P-SUM
5.0 Terms and Abbreviations	Y
Appendix A: Detailed Call Hierarchy	Y

Notes:

Y Included and CompleteSUMSoftware User's Manual

N Not IncludedSPMSoftware Programmer's Manual

P Partial Treatment

Y-XXX Complete, but in Another Manual

P-XXX Partial Treatment in Another Manual

5.1.3 Completeness of Software Analyst's Manual

The information required for the SAM for ALARM Version 3.1 is contained in the Operational Concepts Document (SAM) [Ref. 6]. It is close to fulfilling all the requirements for a standard SAM; however, several topics were either not discussed at all, discussed inadequately, or scattered among different sections and not grouped together. The most significant omissions were those regarding discussions of assumptions and limitations and the overall modeling methodology. Other omissions include the lack of discussions on inputs to areas of functionality and on module correlation with functionality.

A subsection for each Functional Element (FE) of the program should be developed and included in standard SAM, Section 3.3 in adherence to SMART Project requirements. The similarity of model functions can be utilized to produce similar SAM functional

descriptions for easy cross-referencing between model documents. In addition, completeness will be increased as more theories and algorithms are documented through CMS development since CMS design information can be modified for inclusion into the SAM. Table 5.1-3 shows the completeness for each SAM section.

Table 5.1-3 Contents Summary of SAM for ALARM.

Software Analyst's Manual	Complete
Title Page and Preliminary Information	P
1.1 Identification	Y
1.2 System Overview	Y-SAM, SUM
1.3 Document Overview	Y
2.0 Referenced Documents	Y
3.1 Functional Description Overview	P
3.2.1 Assumptions and Limitations	N
3.2.2 Descriptions of Overall Methodology	N
3.3 Detailed Functional Implementation Methodology	
a. Equations and Algorithms	P
b. Equations for Variables	Y
c. Inputs and Outputs	P
d. Module Correlation with Functionality	P
e. Impact on Model Results	Y
4.0 Terms and Abbreviations	Y
Appendix A: Detailed Assumptions and Limitations	N

Notes:

Y Included and CompleteSUMSoftware User's Manual

N Not IncludedSAMSoftware Analyst's Manual

P Partial Treatment

Y-XXX Complete, but in Another Manual

5.2 COMPLIANCE

This section presents ALARM documentation standards and documentation compliance discussions for accomplishing these standards. Information related to the SUM, SPM, and SAM was assessed for compliance using the standards summarized in the following sections. Details of this assessment are presented below.

5.2.1 Software User's Manual

The purpose of the SUM is to provide information and instructions enabling the user to execute a model. It should describe the execution steps, the expected output, and necessary actions when error messages appear. The SUM also provides an introduction to the model. The SUM facilitates the capability to operate the model correctly and to obtain the background for a deeper understanding of the model. The model should be described at a high-level using summarized theoretical information.

5.2.1.1 Standards

The recommended format and contents for a SUM are described in [Ref. 4] and repeated below:

Title Page and Preliminary Information. A SUM Title Page should include the following information: Model Name, Version Number, Volume Number (if applicable), Development Agency, Contract and CDRL Numbers (if applicable), Date Published, Distribution and Destruction notices (if applicable), and Document Control Number (DCN). The contractor name and address should also be included on the Title Page, prefaced by the term "Prepared by." In addition to the Title Page, a Foreword (Abstract), Table of Contents, List of Tables, and List of Figures should be provided as well.

SECTION 1: INTRODUCTION

1.1 Identification. Identify the exact model title, its acronym or abbreviation, the version number, and any other official model identification information.

1.2 System Overview. State the purpose of the model. Include its mission, a general description of the physical systems simulated, and a general description of the intended scenarios. Provide overviews of all major modes of operation and scenarios corresponding to each mode. Auxiliary programs used to generate input data or process output data should be acknowledged and such auxiliary software should be detailed in Section 3.3 (entitled "Links to Other Programs").

1.3 Document Overview. List and describe the purpose of each section of the SUM. Also identify any other documents in the Document Set containing the SUM.

SECTION 2: REFERENCED DOCUMENT

List the title, number, author, publisher, date and classification level (unless all are unclassified) for each document used in generating the SUM and for all known documentation for this model. Include sources for all documents not available through normal government stocking activities.

SECTION 3: EXECUTION PROCEDURES

Present detailed procedures necessary to run the model. The instruction set should be comprehensible by a user unfamiliar with the software design. Each subsection in this section should describe step-by-step instructions for executing the model which includes details of the options available to the user at each step.

3.1 Initialization. Describe the initialization procedures necessary to execute the model. Detail all initialization options.

3.2 User Inputs. Describe user inputs at the file or data set level. Include variable name, format, allowable ranges, units of measure, and definition of each input item.

3.3 Links to Other Programs. Detail model relationships with pre- and post-processors. Describe drivers not considered part of the model but are part of the delivered model package. Discuss any other program with a link to a model.

3.4 Outputs. Detail the expected outputs from the model. This includes narrative reports as well as files. When applicable, give filenames with paths, data format, and units of measure.

SECTION 4: ERROR MESSAGES

List each possible error message with a detailed explanation of each message. Provide a definitive course of action for each error message which includes instructions for restarting the model.

SECTION 5: NOTES

5.1 Glossary of Terms.

5.2 Abbreviations.

APPENDICES

Appendices may be used for ease in document maintenance or for readability of the core text material. Examples of appendix contents are graphs, sample user interface printouts, and any classified information.

APPENDIX A: DETAILED ASSUMPTIONS AND LIMITATIONS

Appendix A is reserved for describing all model assumptions and limitations. These should be organized by major areas of functionality.

5.2.1.2 Assessment

The ALARM Version 3.1 SUM [Ref. 4] was found to provide nearly all the information required for a standard SUM. Most of the contents of the recommended standard SUM can be copied directly from the existing ALARM 3.1 SUM because the section sequences are almost identical. A detailed assessment of the ALARM 3.1 SUM is described in the paragraphs which follow. These contain specific recommendations for satisfaction of the SUM requirements listed in Table 5.3-1

Title Page and Preliminary Information. The Title Page, Foreword, and Table of Contents comply with the requirements of the recommended standard SUM. However, no Lists of Tables and Figures exist in the current SUM and need to be developed for the standard SUM.

1.0 INTRODUCTION.

1.1 Identification. The current SUM includes a good identification of ALARM in Section 1.1. However, the discussion of off-line tools should be moved to the system overview in Section 1.2 of the standard SUM, SPM, and SAM. The remaining information should stay in standard SUM, Section 1.1.

1.2 System Overview. The ALARM SUM includes a good top-level functional overview of the simulation in Section 1.2. This section contains a brief summary of the general areas modeled by the software. This overview should be included in the standard SUM, Section 1.2. The high-level discussion of the off-line support tools in current SUM, Section 1.1 should be moved to standard SUM, Section 1.2. In addition, the ALARM Version 3.1 description in current SUM, Section 3.2 and the Analysis Notes in current SUM, Section 3.6 contain system overview information which should be incorporated into standard SUM, Section 1.2.

1.3 Document Overview. The document overview in current SUM, Section 1.3 complies with the recommended documentation standards. However, modifications may be required to accommodate the recommended changes that are incorporated into the standard SUM.

2.0 REFERENCED DOCUMENTS.

The list of referenced documents in current SUM, Section 2.0 complies with the recommended documentation standards.

3.0 EXECUTION PROCEDURES.

3.1 Initialization. ALARM initialization procedures are covered in Section 3.2.1 of the current SUM. This section explains how to run ALARM along with some of it's basic operating procedures. In addition, a discussion on UNIX compatibility is in Section 3.2.5 of the current SUM. This section covers program compilation in both the VAX and UNIX environments. The information from these sections should be included as subsections of standard SUM, Section 3.1. The title of the subsection currently titled "Initialization" should be renamed to be "Program Execution" since the former title is the name of the main section.

3.2 User Inputs. A discussion on user inputs is covered in current SUM, Section 3.2.2, while that on system inputs is in Section 3.2.3. The section on system inputs discusses inputs from the DMABIO program for scenario mapping. The information from these sections should be included as subsections to standard SUM, Section 3.2. The title of the subsection currently titled "User Input" should be renamed to be "Inputs" since the former title is the name of the main section. The information on Decreasing Model Run Time in current SUM, Section 3.5 covers input setup techniques and should also be included in standard SUM, Section 3.2. In addition, a high-level discussion of input files and the datablocks that comprise them should be developed for inclusion into the standard SUM, Section 3.2.

3.3 Links To Other Programs. ALARM Version 3.1 is supported by eight other software packages for pre- and post-processing functions. DMABIO is a pre-processing package used to generate Defense Mapping Agency (DMA) maps for ALARM program execution. This package is described in detail in current SUM, Section 3.1. The remaining seven packages are described in Appendix F of the current SUM. Four of these tools are used for binary plot file post-processing: PDMERG, BINPRO, and PREPGP (or PREPXP). The PREPGP program is used to prepare ALARM output data for plotting with the GNUPLOT software package. References to a similar program, PREPXP, are also made in the discussion of PREPGP; however, this program is not discussed in its own section. PREPXP is a program for preparing ALARM output data for

plotting with the XPRISM2 and XPRISM3 routines of University of New Mexico's (UNM) Khoros software package. The other three support tools, GENANT, GRAPHIT, and DIMENS, are used for the generation of antenna patterns, the generation of line plots from program outputs when in flight path mode, and for the dimensioning of program parameters, respectively. The support tools GENANT and DIMENS are pre-processing routines while the GRAPHIT support tool is a post-processing routine. The program information in both of these sections should be consolidated in standard SUM, Section 3.3.

3.4 Outputs. These are included in the discussion on model output data in current SUM, Section 3.2.4. These descriptions should be included in standard SUM, Section 3.4. In addition, a high-level discussion of output files should also be developed for inclusion in standard SUM, Section 3.4.

4.0 ERROR MESSAGES.

General explanations are given about how to diagnose error messages; but, no examples are described. A list of error messages is not included in the current documentation. Instead, it states that error messages will be in the output file, are self-explanatory, and will refer the user to the appropriate section of the input data set. No suggestions are provided. These error message discussions are contained in current SUM, Section 4.0. The documentation needs to elaborate on all error messages that can occur and also on subsequent corrective actions. This additional information should be included with the current writeup in standard SUM, Section 4.0.

5.0 TERMS AND ABBREVIATIONS.

A list of Terms and Abbreviations is available in Section 5.0 of the current SUM. This section complies with the recommended standards for a SUM.

APPENDIX A: DETAILED ASSUMPTIONS AND LIMITATIONS.

The specific discussion of model assumptions and limitations was not found in the current documentation; however, references to them are found in scattered sections throughout the current SAM. Several of the more significant limitations are discussed in the system overview in current SUM, Section 1.2 A summarization of program assumptions and limitations should be developed for each major area of functionality and included in standard SUM, Appendix A.

Other Appendices. Since it is recommended that Appendix A of the standard SUM be reserved for Detailed Assumptions and Limitations, the appendices of the current SUM need to be moved. Move the ALARM 3.1 Input Guide and Output Formats information found in current SUM, Appendices A and B to standard SUM, Appendices B and C, respectively. Current Appendix C containing sample ALARM 3.1 input files should be moved to Appendix D, while Appendix D for the output files should be moved to Appendix E. The sample output files in current Appendix D lack discussions on their contents. A short discussion of each output example in this appendix should be developed for inclusion into Appendix E of the recommended SUM. In addition, current Appendix E containing the Run Preparation Instructions should be moved to Appendix F. Current Appendix G containing the ALARM 3.1 Change/Error Notification Form can remain in this appendix since current Appendix F is being moved to standard SUM, Section 3.3.

The material covered in Section 3.3 of the current SUM on Changes for ALARM 3.1 is beyond the scope of the recommended SUM standards. This material has historically

been included in a Version Description Document (VDD) as described in the guidelines outlined in DOD-STD-2167A. More recently, information of this type has also been included in the Release Notes (RN) which accompany new software releases. This section should be removed from the standard SUM and included in one of these documents. In addition, the material covered in Section 3.4 of the current SUM on Algorithm Verification is more appropriate for inclusion in the standard SAM. This section should also be removed from the SUM, renamed as "Algorithm Cross-Reference," and included in the standard SAM as Appendix C.

5.2.2 Software Programmer's Manual

The purpose of the SPM is to enable a user or programmer to understand the operation of a model; install, maintain, and modify it; and convert it for use on other computer systems. The SPM addresses the software implementation of the model rather than theoretical considerations. It provides a guide to the internal workings of the software. It includes information on compiling and linking the code as well as descriptions of hardware and software requirements and peculiarities. If hardware or software listed in an SPM is commercially available, its existing documentation should be referenced by document title and number, and the manufacturer should be cited.

5.2.2.1 Standards

The recommended format for a SPM is described in [Ref. 5] and repeated below:

> **Title Page and Preliminary Information.** The SPM Title Page should include the following information: Model Name, Version Number, Volume Number (if applicable), Development Agency, Contract and CDRL Numbers (if applicable), Date Published, Distribution and Destruction Notices (if applicable), and Document Control Number (DCN). The contractor name and address should also be included on the Title Page, prefaced by the term "Prepared by." In addition to the Title Page, a Foreword (Abstract), Table of Contents, List of Tables, and List of Figures should be provided as well.
>
> **SECTION 1: INTRODUCTION**
>
> **1.1 Identification.** Identify the exact model title, its acronym or abbreviation, the version number, and any other official model identification information.
>
> **1.2 System Overview**. State the purpose of the model. Include its mission, a general description of the physical systems simulated, and a general description of the intended scenarios. Provide overviews of all major modes of operation and scenarios corresponding to each mode. Auxiliary programs used to generate input data or process output data should be acknowledged and described.

1.3 Document Overview. List and describe the purpose of each section of the SPM. Also identify any other documents in the document set containing the SPM.

SECTION 2: REFERENCED DOCUMENTS

List the title, number, author, publisher, date, and classification level (unless all are unclassified) for each document used in generating the SPM and for all known documentation for this model. Include sources for all documents not available through normal government stocking activities.

SECTION 3: PROGRAMMING ENVIRONMENT

3.1 Equipment Configuration. Describe the computing devices and operating systems that the model operates on and under (developmental and target environment). List other software required for model execution. An example of a software requirement is a graphical user interface (GUI).

3.2 Operational Information. Describe hardware/operating system characteristics and capabilities required for the model. This includes details such as storage space for the source code with a complete input set, memory requirements with utilization examples, memory protection features, and input/output (I/O) characteristics.

3.3 Compiling and Linking Instructions. Present instructions on compiling and linking the model software, and describe equipment needed for such procedures. Detail applicable names and version numbers of equipment or software.

SECTION 4: PROGRAMMING INFORMATION

4.1 Introduction. Describe in general the applicable programming conventions and style used to develop the model. A short development history emphasizing programming style and convention evolution could be helpful for mature models with a diverse history.

4.2 Call Hierarchy. Present a top-level subroutine (function) call tree. It should branch down only as far as the main routines for each major area of functionality. A comprehensive call hierarchy (probably generated by an automated software tool) should be included in Appendix A.

4.3 Dictionary of Variables. List all variables alphabetically and provide a definition of each (with units of measure). State whether each variable is global or local. If global, give the name of the common block containing it. If local, list the module(s) containing it.

4.4 Global Variables. Global variables are contained in common blocks for programs written in FORTRAN and are called external variables for programs written in C. Other programming languages will have their own conventions for the handling of global variables. Using the convention appropriate to the programming language, list these variables alphabetically. For example, the common blocks from FORTRAN programs should be listed alphabetically. For each block, list the variables contained in it, give a general description of these variables, and list the modules in which it appears. For programs written in other languages, just list the variables alphabetically, give a general description of these variables, and list the modules in which they appear.

4.5 Program, Subroutine, and Function Descriptions. Provide detailed information about each program, subroutine, or function (hereafter called “module”). List modules alphabetically. Library functions should be listed but only briefly described. All other module

descriptions should contain the following information in a clear, concise format useful to a programmer tasked with maintaining the model.

a. Give a brief narrative description of the module, its objective, and method for fulfilling the objective should be stated.

b. Give its location in a specified file, its call sequence, security classification level, and size (number of lines of executable code).

c. Provide a list of calls made by the module and calls to the module.

d. Alphabetically list all variables used by the module. For each variable, list its dimension, type, usage as input and/or output, engineering units, a very brief description, and its usage as an argument, local, or common variable. The user can refer to the Dictionary of Variables (Section 4.3) for a detailed description.

e. Detailed Description. Elaborate on the objectives and methods used to fulfill the objectives stated in the brief description in list item "a" above. Provide a reference in the SAM if a theoretical discussion related to the modeled processes is provided.

4.6 Error Detection and Diagnostic Features. Describe model error diagnostics. Provide a table listing each error condition, the routine(s) in which it is utilized, the model variable(s) involved, and the conditions (logic) causing the error. (These diagnostics also are summarized in the SUM, Section 4).

SECTION 5: NOTES

5.1 Glossary of Terms.

5.2 Abbreviations.

APPENDICES. Appendices may be used for ease in document maintenance or for readability of the core text material. Examples of appendix contents are subroutine call tree, flow diagrams, sample user interface printouts and any classified information.

APPENDIX A: DETAILED CALL HIERARCHY

Present the complete calling hierarchy in this appendix.

5.2.2.2 Assessment

Information on the software implementation of ALARM Version 3.1 is contained in the SPM [Ref. 5]. The current SPM satisfies most of the requirements for a standard SPM. Some of the topics included are described in adequate detail; yet, others have not been addressed. Information missing from the current SPM includes a high-level call hierarchy, and lists of variables, common blocks, and error messages. The following paragraphs contain comments regarding the SPM requirements described in Table 5-3.

Title Page and Preliminary Information. The Title Page, Foreword, and Table of Contents comply with the requirements of the recommended standard SPM. However, no Lists of Tables and Figures exist in the current SPM and need to be developed for the standard SPM.

1.0 INTRODUCTION.

1.1 Identification. Section 1.1 of the current SPM contains ALARM identification information which should be included in standard SPM, Section 1.1. The information on host computer systems in the second sentence of this section should be removed and included in the equipment configuration discussion of standard SPM, Section 3.1.

1.2 System Overview. The system overview in Section 1.2 of the current SPM should be included in standard SPM, Section 1.2. The high-level discussion of the off-line support tools described in current SUM, Section 1.1 should be included in standard SPM, Section 1.2.

1.3 Document Overview. Section 1.3 of the current SPM contains a document overview for this manual. The discussion of the Document Set containing the SPM mentions the SAM but not the SUM. A reference to the SUM needs to be added to the Document Set discussion for inclusion into standard SPM, Section 1.3. In addition, the document overview may require modification for the recommended changes that are incorporated into the standard SPM.

2.0 REFERENCED DOCUMENTS.

The list of referenced documents in current SPM, Section 2.0 complies with recommended documentation standards.

3.0 PROGRAMMING ENVIRONMENT.

3.1 Equipment Configuration. A reference to host computer systems is made in Section 1.1 of the current SPM. ALARM has been hosted on VAX/VMS, VAX/ULTRIX, and SUN/UNIX systems as well as on IBM-compatible PCs. Elaborate on the hardware and software requirements for these computer systems and include them in standard SPM, Section 3.1. The remaining information in current SPM, Section 3.1 should be included in standard SPM, Section 3.1.

3.2 Operational Information. A discussion of disk storage requirements for VAX/VMS systems was found in current SPM, Section 3.1. Remove them from that section and include them in standard SPM, Section 3.2. There are no references to memory requirements found in the Document Set. These should be developed for inclusion into standard SPM, Section 3.2. Capabilities for VAX/ULTRIX, SUN/UNIX, and PC/DOS systems should be added to standard SPM, Section 3.2 as well, provided they differ from those of VAX/VMS systems. Add performance statistics for VAX/ULTRIX and PC/DOS systems to Table 3.2-1 in current SPM, Section 3.2 for inclusion into standard SPM, Section 3.2. The information in current SPM, Section 3.2 should be included in standard SPM, Section 3.2.

3.3 Compiling and Linking Instructions. Software compilation and linking information is provided in current SPM, Section 3.3. Subroutines needed for compilation on VMS and UNIX systems are listed as well. All this information should be copied to standard SPM, Section 3.3. The subroutines required for compilation on IBM-compatible PCs should be listed for inclusion in standard SPM, Section 3.3 as well. A command file for compiling and linking is

provided with the ALARM installation package. A printout of this command file should be reproduced in Appendix B of the proposed SPM.

Additional requirements for equipment and software do not currently exist. However, ALARM 3.1 is delivered with eight support routines, several of which require separate plotting packages, as described in current SUM, Section 3.1 and Appendix F. No mention of these programs is made anywhere in the SPM. Three of these programs (PREPGP, PREPXP, and GRAPHIT) are post-processing plot routines that use the GNUPLOT plotting package. Instructions on the compiling and linking of these eight programs would be very useful, especially for the plot routines since they are notoriously sensitive to system and device requirements. This information should be included in standard SPM, Section 3.3. The SUM mentions that separate command files exist to build executables for these programs in both the VMS and UNIX environments; but, doesn't provide any detailed information on these files. Printouts of these command files should be included with that for ALARM in Appendix B of the proposed SPM.

4.0 PROGRAMMING INFORMATION.

4.1 Introduction. The first paragraph of current SPM, Section 4.0 should be included in standard SPM, Section 4.1. The remainder of Section 4.0 is a description of ALARM's main modules which belongs with the high-level call hierarchy of standard SPM, Section 4.2. Descriptions of programming conventions and styles should be developed for standard SPM, Section 4.1. A short developmental history emphasizing programming style and convention evolution should be described in this section as well.

4.2 Call Hierarchy. Definitions of the major areas of functionality are given in current SPM, Section 4.0. A high-level call tree consisting of the top two levels illustrated in the flow tree of Appendix A of the current SPM should be developed and included with these definitions in standard SPM, Section 4.2. The detailed call tree in Appendix A of the current SPM should remain in Appendix A for the standard SPM.

4.3 Dictionary of Variables. None of the current manuals contain a variable dictionary; although, variables for input file datablocks are described in Appendix A of the current SUM. While these variables correlate to those contained in the source code, a complete alphabetical list of the variables contained in all program modules needs to be developed to include definitions and units of measure for standard SPM, Section 4.3.

4.4 Global Variables. Global variables are contained in common blocks for programs written in the FORTRAN language. ALARM Version 3.1 is written in FORTRAN according to FORTRAN 77 standards. Discussions of common blocks have not been made except for a statement that they are all incorporated directly into the code. This section should define all common blocks, what routines they are used in, what the variables are, and how to make variable changes. The utility routine "DIMENS" which is used to change parameter values throughout the source code should also be described. All of this information should be included in Section 4.4 of the standard SPM.

4.5 Program, Subroutine, and Function Descriptions. The bulk of the SPM is devoted to this topic. High-level subroutine descriptions are provided in Sections 4.1 through 4.6 of the current SPM while brief descriptions of all program subroutines are included in Appendix B of the current SPM. Descriptions of all program subroutines should be developed according to the SPM standards shown in Section 5.2.2.1 and in Figure 5-1. The recommended approach would be to develop these descriptions using the subroutines in Appendix B of the current SPM and then incorporating the high-level subroutine descriptions from Sections 4.2 through 4.6 into their corresponding modules. The resulting information should be included in Section 4.5 of the standard SPM.

4.6 Error Detection and Diagnostic Features. There are no specific error diagnostics mentioned in the SPM, just generic descriptions of the error checking routines. General explanations are given about how to diagnose error messages in current SUM, Section 4.0; but, no examples are described. A list of error messages is not included anywhere in the current documentation. Instead, it states that error messages will be in the output file, are self-explanatory, and will refer the user to the appropriate section of the input data set. No suggestions are provided. The documentation needs to elaborate on all error messages that can occur and also on subsequent corrective actions. Detailed descriptions of errors and conditions should be included in standard SPM, Section 4.6. These diagnostics should be similar to those recommended for standard SUM, Section 4.0.

5.0 TERMS AND ABBREVIATIONS.

The list of Terms and Abbreviations in current SPM, Section 5.0 complies with recommended documentation standards.

APPENDIX A: DETAILED CALL HIERARCHY.

The detailed call hierarchy in standard SPM, Appendix A complies with the recommended documentation standards.

Other Appendices. The description of ALARM Version 3.1 modules in current SPM, Appendix B should be expanded to include location, sequence, size, and variable descriptions. This information should be removed from Appendix B and integrated with the high-level module descriptions in current SPM, Sections 4.1 through 4.6 for inclusion in standard SPM, Section 4.5. Printouts of command files for the linking and compilation of ALARM 3.1 and the eight off-line support programs should be included in SPM, Appendix B.

FUNCTION NAME: RADVEL

1. Brief Description: Computes the radial velocity of an object with respect to the radar.

2. Calling Sequence: X = RADVEL (P,RANGE)

3. Security Classification: Unclassified

4. Program Size: 8 lines

5. Location: File RAD1.FOR, Line 876

6. Calling Environment

   Calls: DOT
   Called By: ENDRUN, HITPRB, PERCUE, SIGNL, SRCH1, SRCH2

7. Common Blocks: WLTRK/WLTRK, REFLEL/REFLEL

8. Variables:

Name	Usage	Dim	I/O	Type	Definition	Units
P	arg	3,3	x	R	pos, vel, acc of object	m, m/sec
RANGE	arg	1	x	R	range to object	m/sec2
SWIND	com	1	x	R	wind speed against object	m
XDUM	loc	1	---	R	dummy variable	m/s
K	loc	1	---	I	do-loop index	---

9. Discussion and Formulation:

The position vector P of the object is stored in P(1,1), P(2,1) and P(3,1), while the velocity vector V of the object is in P(1,2), P(2,2) and P(3,2). The radial velocity of the object is:

$$RADVAL = \frac{\vec{V} \cdot \vec{P}}{R}$$ where R is the range to object (5.2-1)

Figure 5.2-1 Example of Summary Subroutine Description.

5.2.3 Software Analyst's Manual

The purpose of the SAM is to describe the functional structure and algorithms of a model. It should describe the purpose and background of the model in general terms and give detailed technical descriptions of its complete capabilities, structure, and functions. These detailed descriptions should divide the capabilities of the model into the major functions it performs. All equations, algorithms, and decision processes used by each major function should be described in detail. Details also should be given about model assumptions, limitations, and flexibility (ability to address different types of problems). Inputs and outputs should be described in words rather than file formats. Each module should be described in great detail to explain the correlation between the modules and model functional descriptions. The SAM enables the user to understand the theoretical basis of the model. The user needs it to facilitate understanding of the code and to ensure that the model is appropriate for particular analysis requirements.

5.2.3.1 Standards

The recommended format for a SAM is described in [Ref. 6] and repeated below:

Title Page and Preliminary Information. The SAM Title Page should include the following information: Model Name, Version Number, Volume Number (if applicable), Development Agency, Contract and CDRL Numbers (if applicable), Date Published, Distribution and Destruction Notices (if applicable), and Document Control Number (DCN). The contractor name and address should also be included on the Title Page, prefaced by the term "Prepared by." In addition to the Title Page, a Foreword (Abstract), Table of Contents, List of Tables, and List of Figures should be provided as well.

SECTION 1: INTRODUCTION

1.1 Identification. Identify the exact model title, its acronym or abbreviation, the version number, and any other official model identification information.

1.2 System Overview. State the purpose of the model. Include its mission, a general description of the physical systems simulated, and a general description of the intended scenarios. Discuss the types of problems addressed and types of answers provided by the model. Provide overviews of all major modes of operation and scenarios corresponding to each mode. Auxiliary programs used to generate input data or process output data should be acknowledged and described.

1.3 Document Overview. List and describe the purpose of each section of the SAM. Also identify any other documents in the document set containing the SAM.

SECTION 2: REFERENCED DOCUMENTS

List the title, number, author, publisher, date, and classification level (unless all are unclassified) for each document used in generating the SAM and for all known documentation for this model. Include sources for all documents not available through normal government stocking activities.

SECTION 3: FUNCTIONAL DESCRIPTION

3.1 Overview.

Describe the model's complete functionality without reference to implementation methodology. These descriptions should elaborate on the overall mission and major modes described above in System Overview, Section 1.2. Descriptions should be presented in the order in which detailed functional methodologies are described in the sections that follow.

3.2 General Modeling Approach

3.2.1 Assumptions and Limitations. Describe high-level assumptions and limitations of overall model functionality.

3.2.2 Overall Modeling Methodology. Explain how assumptions, limitations, and the processes involved influence the general modeling methodology.

3.3 Detailed Functional Implementation Methodology

Describe how the capabilities of the model are functionally implemented. Divide this section into subsections corresponding to the model's major areas of functionality and provide the following information for each subsection:

a. **Equations and Algorithms.** Provide detailed technical descriptions and purposes for use of specific empirical and analytic equations, numerical algorithms, and decision processes used by the function. Use flow diagrams to depict the implemented logic and use illustrations to depict geometrical considerations when appropriate. Justify use of specific probability distributions. When trade-off studies for equation usage were performed, justify use of the chosen equation.

b. **Equations for Variables.** Present and describe all equations (using mathematical notation) used for calculating variables that are significant in the implementation of the functionality. Indicate the code variable names that correspond with the variables described by these equations.

c. **Model Inputs and Outputs.** Inputs and outputs relevant to a particular area of functionality should be described in words without reference to code implementation details. Identify the relationship of inputs to the equations and algorithms in one of those areas.

d. **Code Module Correlation with Functionality.** Identify each module used to implement an area of functionality and describe the processes contained in that module. The description of each module should include its purpose, a detailed technical explanation, and definition of variables. Correlate these processes with the model functional descriptions. Applicable library functions may simply be listed with a short description.

e. **Impact on Model Results.** Describe the impact of the functionality on model results.

SECTION 4: NOTES

4.1 Glossary of Terms.

4.2 Abbreviations.

APPENDICES. Appendices may be used for ease in document maintenance, examples and illustrations to assist in understanding model capabilities, or for readability of the core text material. Examples of appendix contents are logic flow diagrams, sample user interface printouts, examples of post-processor use, former studies published using this model, and any classified appendices.

APPENDIX A: DETAILED ASSUMPTIONS AND LIMITATIONS

Appendix A is reserved for describing all model assumptions and limitations. These should be organized by major areas of functionality. This appendix is the same as Appendix A of the SUM.

5.2.3.2 Assessment

Information on the engineering implementation of ALARM Version 3.1 is contained in a document entitled Operational Concepts Document (Analyst's Manual) for the Advanced Low Altitude Radar Model or the SAM [Ref. 6]. It is close to fulfilling all the requirements for a standard SAM. Most of the topics included are described in adequate detail; but, some topics have not been addressed. The following paragraphs contain comments regarding the SAM requirements described in Table 5-4.

Title Page and Preliminary Information. The Title Page, Foreword, and Table of Contents comply with the requirements of the recommended standard SAM. However, no Lists of Tables and Figures exist in the current SAM and need to be developed for the standard SAM.

1.0 INTRODUCTION.

1.1 Identification. Section 1.1 of the SAM contains ALARM identification information that should be included in standard SAM, Section 1.1.

1.2 System Overview. The purpose of this software model is discussed in Section 1.2 of the SAM. This discussion should be included in standard SAM, Section 1.2 which should be renamed as "System Overview". The overview in current SAM, Section 4.1 should be included into standard SAM, Section 1.2 (except for the last paragraph). A functional description (system architecture) overview should be developed instead for its former section. In addition, the discussion of the off-line post-processing tools in current SUM, Section 1.1 should also be included

in standard SAM, Section 1.2. Finally, the model mission discussions in Sections 3.0 through 3.3 of the current SAM should be integrated into standard SAM, Section 1.2.

1.3 Document Overview. The document overview is covered in the Introduction in current SAM, Section 1.3. This section needs to be renamed as the "Document Overview"; otherwise, the discussion complies with recommendations for standard SAM, Section 1.3. However, modifications may be required for the recommended changes that are incorporated into the standard SAM.

2.0 REFERENCED DOCUMENTS.

The list of referenced documents in Section 2.0 of the current SAM complies with the recommended documentation standards.

3.0 FUNCTIONAL DESCRIPTION.

3.1 Overview. This topic is partially covered in the last paragraph of the overview in current SAM, Section 4.1 and should be included into standard SAM, Section 3.1. Develop a more detailed functional description (system architecture) overview elaborating on the overall mission and major modes presented in the system overview for inclusion into standard SAM, Section 3.1.

3.2 General Modeling Approach

3.2.1 Assumptions and Limitations. Discussions on these topics are scattered throughout the manual. Several of the more significant limitations are discussed in the overview in current SAM, Section 4.1. A discussion of high-level assumptions and limitations should be developed for inclusion into standard SAM, Section 3.2.1. All assumptions and limitations should be discussed in detail and included in Appendix A of the standard SAM (same as Appendix A in the standard SUM).

3.2.2 Overall Modeling Methodology. A high-level methodology description has not been developed for ALARM. Some of these topics are scattered throughout the SAM. A section explaining how assumptions, limitations, and the processes involved influence the general modeling methodology should be generated for standard SAM, Section 3.2.2.

3.3 Detailed Functional Implementation Methodology. Doppler processing is treated in the current SAM; but, the only formulas listed are for a Chebychev filter and for integrating the response at several distinct pulse repetition frequencies (PRFs). Specific information on coherent integration should be developed to address subparagraphs a. through e. in Section 3.3 of the SAM standards. This information should be incorporated in the standard SAM, Section 3.3. Page 94 of the FET/DRD/FED includes a brief description of Doppler processing that would be a good introduction to this topic.

a. **Equations and Algorithms.** Adequate descriptions of equations and algorithms are covered in Sections 4.2 through 4.5 of the current SAM. Justifications for probability density functions used in probability of detection (Pd) and false alarm (Pfa) calculations in current SAM, Section 4.4.6 are also adequate. No empirical functions were found in the existing documentation. All of the information in these sections should be included in standard SAM, Section 3.3.

b. **Equations for Variables.** Adequate descriptions of the equations used to calculate variable values for the main program equations and algorithms cited are provided in Sections 4.2 through 4.5 of the current SUM. These descriptions should be included in SAM, Section 3.3 along with the equations and algorithms for which they calculate variable values.

c. **Inputs and Outputs.** Descriptions of each main area of functionality in the program are included in current SAM, Sections 4.2 through 4.5. While the outputs returned from areas of functionality regarding external and internal signals are provided, no description of the inputs is given. Inputs to and outputs returned from the other areas of functionality aren't clearly defined. Tables detailing the relationship of inputs and outputs to each area of functionality listed in these sections should be developed for inclusion into standard SAM, Section 3.3. These tables should be included with the discussions for each area of functionality covered in current SAM, Sections 4.2 through 4.5.

d. **Module Correlation with Functionality.** Mid-level modules used to implement areas of functionality are referenced throughout Sections 4.2 through 4.5 of the current SAM. However, some of the lower-level modules accessed by these mid-level modules aren't discussed in these sections. Further elaboration on the contents of these modules and of the lower-level modules referenced by them is recommended. When reviewing equations from a particular area of functionality, it is not always obvious in which module an equation is found. The source module should be given whenever an equation or variable is listed. These explanations should be included with the content of current SAM, Sections 4.2 through 4.5 in standard SAM, Section 3.3.

e. **Impact on Model Results.** The effects that particular areas of functionality have on model results are discussed in adequate detail in current SAM, Sections 4.2 through 4.5. The information in these sections should be included in standard SAM, Section 3.3.

4.0 TERMS AND ABBREVIATIONS.

A glossary of terms and abbreviations was found in Section 5.0 of the current SAM. This information should be included in standard SAM, Section 4.0 under the title "Terms and Abbreviations" rather than "Glossary."

APPENDIX A: DETAILED ASSUMPTIONS AND LIMITATIONS.

As mentioned above, discussions on these topics are scattered throughout the manual. All assumptions and limitations should be discussed in detail and included in Appendix A of the standard SAM (same as Appendix A in the standard SUM). A discussion of high-level assumptions and limitations should be included in standard SAM, Section 3.2.1.

Other Appendices. Since standard SAM, Appendix A is reserved for describing assumptions and limitations, the Clutter Reflectivity figures in Appendix A of the current SAM should be moved to Appendix B of the proposed SAM. In addition, the algorithm verification discussion in current SUM, Section 3.4 covers material that is more appropriate for inclusion to the SAM. This section should be renamed as "Algorithm Cross-Reference" and moved to Appendix C of the standard SAM.

5.3 RECOMMENDED MODIFICATIONS

The sections that follow describe the changes needed to bring ALARM Version 3.1 documentation into compliance with the standards recommended in [Ref. 9]. Table entries provide estimates of the number of additional pages (based on the current manuals page count) needed to complete such recommendations. Comments are also included in these tables regarding what recommendations are being made. These estimates are rough order of magnitude (ROM) based on the current understanding of ALARM and its documentation. Wherever possible, the page estimates are based on treatments of similar topics in the ALARM documentation or in the documentation of other models.

5.3.1 ALARM Software User's Manual

The current SUM is very close to fulfilling all the requirements for a standard SUM. Table 5.3-1 presents a summary of the recommendations from the above discussions to bring the SUM into compliance with the proposed documentation standards.

.

Table 5.3-1 Estimated Number of New Pages for ALARM SUM

Section/Topic	Number of New Pages	Recommendations
Title Page and Preliminary Information	1	Develop lists of all figures and tables contained in the SUM.
1.1 Identification	0	Copy from current SUM, Section 1.1. Put discussion of off-line tools in standard SUM, Section 1.2.
1.2 System Overview	0	Copy from current SUM, Sections 1.2 and 3.6. Copy discussion of off-line tools from current SUM, Section 1.1.
1.3 Document Overview	0	Copy from current SUM, Section 1.3.
2.0 Referenced Documents	0	Copy from current SUM, Section 2.0.
3.1 Initialization	0	Copy from current SUM, Sections 3.2.1 and 3.2.5.
3.2 User Inputs	1	Copy from current SUM, Sections 3.2.2, 3.2.3, and 3.5. Develop high-level discussion of input file formats.
3.3 Links to Other Programs	0	Copy from Section 3.1 and Appendix F of current SUM.
3.4 Output Data	1	Copy from Section 3.1.4. Develop high-level discussion of output file formats.
4.0 Error Messages	5	Copy from current SUM, Section 4.0. Develop detailed error message/action descriptions (most of this section will be new).
5.0 Terms and Abbreviations	0	Copy from current SUM, Section 5.0.
Appendix A: Detailed Assumptionsand Limitations	10	All new and should be developed in detail for standard SUM, Appendix A.

Table 5.3-1 Estimated Number of New Pages for ALARM SUM

Section/Topic	Number of New Pages	Recommendations
Other Appendices	5	Move current SUM Appendices A to B, B to C, C to D, D to E, and E to F. Also, briefly describe sample output files in Appendix D of current SUM.

Additional recommendations not covered in the above discussions are as follows:

The discussion on changes for ALARM Version 3.1 in current SUM, Section 3.3 doesn't belong in this manual and should be removed. It should be put in either a Release Notes (RN) document or a Version Description Document (VDD). The discussion concerning Algorithm Verification in current SUM Section 3.4 should be renamed as "Algorithm Cross-Reference" and moved to Appendix C of the proposed SAM.

5.3.2 ALARM Software Programmer's Manual

The current SPM is very close to fulfilling all the requirements for a standard SPM. Table 5.3-2 presents a summary of the recommendations from the above discussions to bring the SPM into compliance with the proposed documentation standards.

Table 5.3-2 Estimated Number of New Pages for ALARM SPM

Section/Topic	Number of New Pages	Recommendations
Title Page and Preliminary Information	1	Develop lists of all figures and tables contained in the SPM.
1.1 Identification	0	Copy from current SPM Section 1.1. Put equipment discussion in standard SPM Section 3.1.
1.2 System Overview	0	Copy from current SPM, Section 1.2. Copy discussion of off-line tools from current SUM, Section 1.1.
1.3 Document Overview	1/3	Copy from current SPM, Section 1.3 and include SUM in Document Set discussion.
2.0 Referenced Documents	0	Copy from current SPM, Section 2.0.
3.1 Equipment Configuration	1/3	Copy from current SPM, Section 3.1. Copy host computer system discussion from current SPM, Section 1.1. Elaborate on the H/W and S/W requirements for these systems. Put disk storage requirements in standard SPM, Section 3.2.
3.2 Operational Information	1/3	Copy from current SPM, Section 3.2. Copy disk storage requirements from current SPM, Section 3.1. Add memory requirements and performance statistics for VAX/ULTRIX and PC/DOS systems.

Table 5.3-2 Estimated Number of New Pages for ALARM SPM

Section/Topic	Number of New Pages	Recommendations
3.3 Compiling and Linking Instructions	1	Copy from current SPM, Section 3.3. Add discussions on the compilation and linking of the off-line support programs.
4.1 Introduction of Programming Information	1	Add section on coding conventions with short developmental history. Copy module definitions to standard SPM, Section 4.2.
4.2 Call Hierarchy	1	Copy top two levels from detailed call tree in current SPM, Appendix A to high-level call tree for standard SPM, Section 4.2. Copy module definitions from current SPM, Section 4.0.
4.3 Dictionary of Variables	20	Develop a variable definition list.
4.4 Global Variables	20	Develop a common block definition list and describe DIMENS routine.
4.5 Program, Subroutine and Function Descriptions	80	Copy from Sections 4.1 through 4.6 and Appendix B of current SPM. Expand module descriptions.
4.6 Error Detection and Diagnostic Features	N/2	Mostly new and should develop error condition and diagnostic table (N = number of error diagnostics).
5.0 Terms and Abbreviations	0	Copy from current SPM, Section 4.0.
Appendix A: Detailed Call Hierarchy	0	Copy from Appendix A of current SPM.
Other Appendices	10	Modify Appendix B of current SPM and copy to standard SPM, Section 4.5. Put command file printouts for the compilation and linking of ALARM and the eight off-line support tools in proposed SPM, Appendix B.

5.3.3 ALARM Software Analyst's Manual

The ALARM Version 3.1 SAM is close to fulfilling all the requirements for a standard SAM. Table 5.3-3 presents a summary of the recommendations from the above discussions to create a standard SAM.

Table 5.3-3 Estimated Number of New Pages for ALARM SAM.

Section/Topic	Number of New Pages	Recommendations
Title Page and Preliminary Information	1	Develop lists of all figures and tables contained in the SAM.
1.1 Identification	0	Copy from current SAM, Section 1.1.
1.2 System Overview	0	Copy from current SAM Sections 1.2, 3.0, and 4.1 (except for last paragraph). Copy discussion of off-line tools from current SUM, Section 1.1 and rename as "System Overview".
1.3 Document Overview	0	Copy from current SAM, Section 1.3 and rename as "Document Overview".

Table 5.3-3 Estimated Number of New Pages for ALARM SAM.

Section/Topic	Number of New Pages	Recommendations
2.0 Referenced Documents	0	Copy from current SAM, Section 2.0.
3.1 Functional Description Overview	1	Copy last paragraph from current SAM, Section 4.1. Mostly new and should be developed.
3.2.1 Assumptions and Limitations	1	Mostly new and some can be copied from scattered sections of current SAM. Describe only high-level assumptions and limitations.
3.2.2 Overall Modeling Methodology	2	All new.
3.3 Detailed Functional Implementation Methodology		Add coherent integration algorithms addressing all topics in Section 3.3 (subparagraphs a. through e.).
a. Equations and Algorithms	X/2	Copy from Sections 4.2 through 4.5 of the current SAM (X = # of undescribed algorithms).
b. Equations for Variables	0	Copy from Sections 4.2 through 4.5 of the current SAM.
c. Inputs and Outputs	Y/2	Develop tables for listing inputs and outputs for areas of functionality (Y = # of areas of functionality).
d. Module Correlation with Functionality	Y/2	Develop list of modules associated with areas of functionality and put them in tables with inputs and outputs (Y = # of areas of functionality).
e. Impact on Model Results	0	Included with algorithm descriptions.
4.0 Terms and Abbreviations	0	Copy from Glossary in current SAM, Section 5.0 and rename as “Terms and Abbreviations”.
Appendix A: Detailed Assumptions andLimitations	10	Mostly new and some can be copied from scattered sections of current SAM.
Other Appendices	0	Copy Appendix A of current SAM to Appendix B of proposed SAM. Copy current SUM, Section 3.4 to proposed SAM, Appendix C and rename as “Algorithm Cross-Reference”.

5.3.4 Summary

A significant level of effort will be required to generate three standard manuals as described in Section 5.2. Most of this effort would go for generating the proposed SPM with substantially less effort required for generation of the proposed SUM and SAM. Table 5.3-4 summarizes the estimated number of new pages required for each manual. A quick comparison of this table with the summary tables in the earlier documentation assessment for ALARM 3.0 indicates that the ALARM 3.1 documentation efforts did not consider the earlier documentation assessment in updating the Documentation Set.

Table 5.3-4 Summary: Estimated Number of New Pages.

Manual	Additional Pages
SUM	23
SPM	135 + N/2
SAM	15 + X/2 + Y
Total	**173 + N/2 + X/2 + Y**

Notes:

N = # of error diagnostics

X = # of algorithms not described

Y = # of areas of functionality

The documentation for ALARM will require much work to fulfill the recommended standards. The missing information will require a fairly large number of new pages. The bulk of the new pages will contain subroutine, variable, and common block descriptions for the proposed SPM. These are not as technically demanding as the theoretical discussions in an SAM; but, time to examine the code and write the descriptions will be required. The rest of the new pages will largely consist of detailed descriptions of assumptions and limitations and error messages/diagnostics. The correlation of modules, inputs and outputs to program areas of functionality, and the discussion of algorithms for coherent integration may also take a significant number of pages.

Model documentation is worth a significant expenditure of resources. The Military Operations Research Society (MORS) has included good documentation as a step in the model validation process [Ref. 6]. Development and use of standard documentation will increase user efficiency as well as model credibility.

5.4 IMPLICATIONS FOR V&V

The quality of the ALARM Version 3.1 documentation is assessed to be good, especially when combined with extensive comments in the source code. The Software User's Manual for Advanced Low Altitude Radar Model (ALARM 3.1) [Ref. 4] identifies and describes all the input data fields and then describes both inputs and results for 16 sample cases provided with the model. The Software Programmer's Manual for Advanced Low Altitude Radar Model (ALARM 3.1) [Ref. 5] details the software implementation of the model to include the description of both the overall structure and each subroutine in the model. The Operational Concepts Manual (Analyst's Manual) for Advanced Low Altitude Radar Model (ALARM 3.1) [Ref. 6] explicitly describes algorithms along with equations and a few references.

No detailed list was found of model errors and corrective actions for the user to take. In addition, there is no comprehensive list of assumptions and limitations for the simulation model. Serious deficiencies include missing algorithms for coherent integration and the lack of the correlation of modules, inputs, and outputs with areas of functionality which is important for V&V activity. COMMON blocks, coding convention, and model variable descriptions are missing from the current documentation which could impede efforts to verify functionality of individual modules.

5.5 IMPLICATIONS FOR MODEL USE

ALARM documentation is generally complete except for SPM-related material in particular. Deficiencies noted in this manual, particularly the lack of variable and COMMON blocks descriptions, can be compensated for by a review of the code. Generally, extensive commenting prior to individual subroutines provide descriptions of all input and output variables used in the routine.

The lack of adequate model execution error diagnostics in the SUM and the SPM could hinder the timely correction of problems. In addition, the absence of a discussion on assumptions and limitations could lead to improper program setup resulting in erroneous interpretation of output data. The description of the output file contents would reduce new-user training efforts.